精度保証付き数値計算について

岩波龍雅（基幹理工学研究科 数学応用数理専攻 修士 2 年）

概要

このポスターでは**精度保証付き数値計算**という，数値計算から数学的に厳密な評価を得る手法を紹介する．

精度保証付き数値計算とは

本節では発表者によるライブラリ Verry[1][2] を用いて，精度保証付き数値計算のコンセプトを説明する．

例として，Lotka–Volterra 方程式

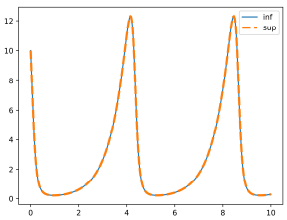
$$\frac{\mathrm{d}x}{\mathrm{d}t} = ax - bxy, \quad \frac{\mathrm{d}y}{\mathrm{d}t} = -cx + dxy$$

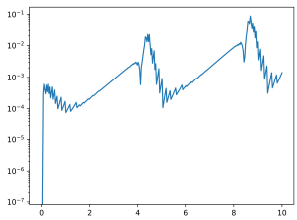
の初期値問題について考える．精度保証付き数値計算では，解の近似値ではなく “包含” を計算する．以下のプログラムは Python で書かれており，同問題を初期値 $(x_0, y_0) = (10, 5)$，パラメータ $(a, b, c, d) = (1.5, -1, 3, 1)$ のもとで $t = 0$ から $t = 10$ まで解く：

```
# !pip install verry
from verry import FloatInterval as FI
from verry.integrate import C0Solver, doubleton, eilo

# 1. 問題の右辺を記述する
def fun(t, x, y):
    return (1.5 * x - x * y, -3 * y + x * y)

# 2. 初期値問題のソルバを定義する
solver = C0Solver(integrator=eilo, tracker=doubleton)

# 3. ソルバを問題に適用する
r = solver.solve(
    fun, t0=FI(0), y0=[FI(10), FI(5)], t_bound=FI(10)
)

assert r.status == "SUCCESS"
x = r.content.y[0]
print(x)  # [inf=0.286776, sup=0.287650] と出力される
```

このプログラムの出力 $[0.286776, 0.287650]$ は，解が時刻 $t = 10$ においてとる値の（数学的に厳密な）下界と上界の組である．

このプログラムは解が $t = 10$ でとる値だけでなく，変域 $[0, 10]$ に属する t すべてについて，解の下界と上界を計算している．





左図は各 t における x の値の下界（青の実線）と上界（橙の破線）を示している．計算された区間 I の直径（**絶対誤差**）は，この変域では視認できないほど小さいことが分かる．右図は各 t について，I の直径を中心で割った値（**相対誤差**）を示している．

次節では，精度保証付き数値計算を支える代表的なテクニックを 3 つ紹介する．

基礎となるテクニック

区間演算

区間演算は区間を対象とする演算であり，演算の前後で包含関係が保たれるように定義される．たとえば，加法は

$$[a, b] + [c, d] = [a + c, b + d]$$

と定義され，定義から

$$\forall x \in [a, b],\ \forall y \in [c, d],\ x + y \in [a, b] + [c, d]$$

が成立する．

区間演算が必要となる理由は 2 つある．第一の理由は，数値計算に使える数（**浮動小数点数**）の個数には限りがあり，表現できない数は近い浮動小数点数に丸められるため，その誤差（**丸め誤差**）を把握しないと厳密な結果が得られないことにある．

第二の理由は，集合（区間）の包含関係を検証できれば，**不動点定理**を用いて問題の解の存在や一意性などを示せることにある．

不動点定理

不動点定理とは，方程式 $x = T(x)$ の解（**不動点**）の存在および個数に関する定理である．以下に代表例を 1 つあげておく：

Banach の不動点定理　T を完備距離空間 $\mathcal{X}$ 上の写像とする．$K \subseteq \mathcal{X}$ における T の Lipschitz 定数が 1 未満のとき T を K 上の**縮小写像**という．閉集合 K 上で T が縮小写像かつ $T(K) \subseteq K$ ならば，T の不動点で K に属するものが一意に存在する．

$x = T(x)$ の解の包含を計算する定番の方法は，T の縮小性を予め（机上で）示してから，適切な閉集合 K について

$$T(K) \subseteq K$$

が成り立つことを計算機で検証するというものである．K は厳密解を含むように，近似解 $\hat{x}$ をわずかに膨らませて作ることが多い．

Newton 法

多くの問題は $x = T(x)$ の形ではなく

$$F(x) = 0$$

の形で定式化される．$T(x) = x - F(x)$ と置けば当然 $x = T(x)$ の形にできるが，$\|T(x) - T(\hat{x})\| \approx \|T'(\hat{x})(x - \hat{x})\|$，$T'(\hat{x}) = I - F'(\hat{x})$ より，T が $\hat{x}$ の近傍で縮小写像になるには $\|I - F'(\hat{x})\| \ll 1$ でなければならない．そこで，$\|I - RF'(\hat{x})\| \ll 1$（i.e., $R \approx F'(\hat{x})^{-1}$）となる単射線形写像 R を計算し，**簡易 Newton 写像**

$$N(x) = x - RF(x)$$

を定める．$\|N(x) - N(\hat{x})\| \approx \|(I - RF'(\hat{x}))(x - \hat{x})\| \ll \|x - \hat{x}\|$ なので，N については $\hat{x}$ の近傍で縮小写像になると期待できる．

参考文献

[1] R. Iwanami, Verry, https://python-verry.github.io/verry/, (17 October 2025).
[2] R. Iwanami, Verry: an open-source package for verified computation written in Python 3, in Proceedings of the 20th International Symposium on Scientific Computing, Computer Arithmetic, and Verified Numerical Computations, Oldenburg, NI, 2025, pp. 18—20, doi:10.5281/zenodo.17584516.